REMARKS

In the Final Office Action dated June 14, 2005, claims 1-22 are stated as pending. Applicants note that claim 22 was cancelled in the Response of March 22, 2005. Claims 1, 5, 8, 9, and 17 are herein amended. Claims 23-25 are newly added. Claim 23 replaces originally filed claim 22. Claims 1, 9, 17, and 23 are independent claims from which all other claims depend therefrom. Applicants note that claims 1 and 9 are herein amended for clarification reasons and that the originally filed and previously presented claims 1 and 9 were in allowable form as presented. Applicants understand that the claims ought to be interpreted in a consistent and agreeable manner in view of the specification. However, Applicants nevertheless provide the amendments of claim 1 and 9 to clarify what is meant by the term "SRML simulation model".

In paragraph 3, the current Office Action states that a declaration was filed May 25, 2001. Applicants note that the declaration was filed March 22, 2005. Regardless of the filing date, Applicants, respectfully, request that the declaration be disregarded.

In paragraphs 4-6, the Office Action states that claim 22 remains rejected under 35 U.S.C. 102(b) as being anticipated by Miller et al., "JSIM: A Java-Based Simulation and Animation Environment", hereinafter Miller '97. Applicants again note that claim 22 was previously cancelled.

In paragraphs 9-13, the Office Action states that claims 1-7 and 9-21 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Miller '97 in view of Huang et al., "Building a Web-Based Federated Simulation System with Jini and XML", hereinafter Huang.

Amended claim 1 recites a simulation system that has a simulation reference simulator. The simulator is adapted to receive an SRML simulation model including a simulation item by way of simulation specific scripts. The simulation reference simulator responds to the simulation specific scripts

included in the model and includes an item manager, for loading properties of each item of the simulation model, and an event manager, for processing the simulation model.

Note that "SRML" stands for Simulation Reference Markup Language, as stated in the specification of the present application. SRML is a markup language that is specific to simulations. Like HTML, SRML uses scripting languages interspersed with markup tags to describe the specific behavior of a document or model. SRML is not specific to a particular type of programming language (i.e. Java), but rather may be used in combination with one or more scripts (e.g. JavaScript) or programming languages that are to be used in performing one or more sets of tasks. A scripting language is an interpreted language that has a set of commands that are executed by a program, and is not a program in and of itself. A scripting language is a language that is loaded and executed dynamically, as opposed to a language that is compiled statically and then executed, such as Java or Visual Basic. Programming languages, such as Java or Visual Basic are language specific and dependent. Scripting language files usually do not "run" or "execute" within their own runtime environment, but rather are executed on behalf of a pre-existing runtime environment.

A scripting language may be extended and has a runtime environment that is dynamic and may be changed. This is unlike a programming language, such as Java or Visual Basic. Once a set of commands is compiled in Java or Visual Basic they are fixed and cannot be modified. The operating environments of Java and Visual Basic are static and cannot be extended. With a scripting language when a set of commands is loaded they may be modified. A scripting is loosely bound or not highly interconnected, whereas, a non-scripted programming language, such as Java and Visual Basic, is highly interconnected. Unlike HTML, SRML models have item class mechanisms, for including the behavior of classes utilizing an inheritance structure.

The claimed invention of claim 1 includes simulator that is adapted to receive an SRML simulation model. A SRML simulation model, as defined in the specification, is a model that may be used to describe a system structure, item behavior, item attributes, processes flow, etc. This behavior description is in the form of scripts, which themselves may refer to behavior provided by one or more programming languages, such as Java or Visual basic for the performance of a set of tasks. Applicants submit that none of the relied upon references provide such a simulator or an SRML simulation model.

Miller '97 discloses a Java-based simulation and animation environment that allows a designer to create simulation models using Java. The creation of simulation models using Java is distinctly different from a system that allows one to use a markup language to create a SRML simulation model. In the first instance one is merely using a single programming language, specifically Java, to create a specific simulation model. In the second instance a markup language is used that may refer to one or more programming languages using scripts to create a SRML simulation model. A SRML simulation model may refer to multiple Java created simulation models, as well as various other models and script-based items, elements, and attributes.

The Office Action states that Miller '97 teaches a simulator adapted to receive a simulation model including a simulation item. Applicants submit that the simulator or simulation model of Miller '97 is completely unlike that of the claimed invention. As stated the simulation model of Miller '97 is a model formed solely from Java programming. The claimed simulator receives the SRML simulation model that has a simulation item by way of scripts. The claimed system provides a SRML simulation model that is not specific to Java and is not formed from the Java programming language, but rather is formed of scripts. Thus, the simulator, the simulation models, and the simulation items are

not the same as that claimed. Note that nowhere in Miller '97 are scripts used to generate a simulation model.

Claim 9 recites a method of conducting simulations that includes defining a simulation model having simulation specific scripts with a Simulation Reference Markup Language (SRML) that is domain specific to simulations. The model is communicated to a simulation reference simulator, which executes the model. The modeled events are outputted.

As with claim 1, Miller '97 fails to teach or suggest a simulation model that has simulation specific scripts, the use of SRML, and a simulator for execution thereof. Not only does Miller '97 fail to teach or suggest the use of SRML, as admitted to in paragraph 11 of the Office Action, Miller '97 also fails to teach or suggest the use of a SRML that is domain specific to simulations or the like. Applicants submit that SRML was originally developed by the Applicants and was disclosed for the first time in the specification of the present application. SRML and the use thereof is novel to the present invention and nothing like it is disclosed in any of the relied upon references.

Claim 17 recites a computer-readable storage medium having computer-readable program code devices embodied therein for simulating events. The medium includes instructions comprising simulation specific scripts representing a simulation model having multiple items. Each of the items represents an article, process, or system and includes a property. The property includes an ItemClass for associating items.

Applicants submit that none of the relied upon references teach or suggest simulation specific scripts or the use of scripts to represent a simulation model.

With respect to claim 18, the Office Action states that Miller '97 teaches an item that includes an ItemID that defines a script object and refers to section 3.4.3 of Miller '97. Applicants, respectfully, traverse. In section 3.4.3, Miller '97 discloses a public variable simId. It appears that the variable simId is used as a

data member to identify a simulation entity. The statement "public long simId ()" is a Java programming language line of code. This statement is not a markup language line of code nor is it a script or a script object. Notice that it does not include tags, like a script.

In paragraph 11, the Office Action states that Miller '97 does not expressly teach the simulation model being an SRML model and an item manager that is in operative communication with a XML parser for building a Document Object Model (DOM) tree in the process of loading items of the simulation model. Applicants agree. However, the items in an SRML simulation model are expressly provided with scripts, which communicate with the DOM, once loaded, and the structure of the classes of items are provided within the markup. Such functionality is not suggested by Miller '97.

In paragraph 13, the Office Action states that since Huang teaches XML based languages that are developed with a common Document Type Definition and methods that combine the XML meta data standard and ontology mapping as an approach to achieve this semantic interoperability, it would have been obvious to create models from languages that are XML-based to enable this interoperability. Applicants traverse.

XML is an extensible markup language that is specific to a generic domain that allows one to define tags and create languages such as HTML and SRML. XML may be used in an infinite number of ways and for an infinite number of purposes. Huang only suggests that simulation data may be converted into an XML format and transported over the web and then may be reconverted from the XML format into class files in an object-oriented programming language, such as Java or C++. This suggestion merely discloses the possibility of being able to manipulate information at an object level instead of an element level. The XML format is merely used as a transport medium for simulation data; it is not used to create simulation models that include their own scripted behavior as

with SRML. The XML is used as a standard format in which simulation data may be transported over the Web and converted into an object-oriented programming language. Upon conversion the portions of the data may be selected, synthesized with other data, stored, and possibly forwarded to other interested parties. Thus, Huang uses XML for a completely different purpose and the mere disclosure of using XML for a specific purpose, namely the transportation of simulation data over the web, does not in anyway suggest the creation of SRML. Although SRML is XML based, the disclosure and use of XML does not suggest the creation of SRML, to suggest otherwise would be the use of improper hindsight reasoning in view of the present application. See pages 145-146, section 3, paragraphs 1-6 and page 148, section 5, first full paragraph of Huang.

The Office Action also states that the XML based languages as taught by Huang have the same functionality as a simulation model created with SMRL since they can be converted to an XML document. Applicants again traverse. Applicants are unable to find anywhere in Huang the conversion of a simulation model that is converted into an XML format, but rather is only able to find the conversion of simulation data into an XML format. Regardless of whether this conversion is disclosed, Huang fails to disclose the creation of a simulation model using a script language. The conversion of a simulation model or of simulation data in a programming language into and out of a XML format for Web based transfer does not provide the functionality of creating a simulation model in SRML. The conversion of programming code and data does not enable the dynamic loading of information to fully describe a simulation nor does it enable the generation of a simulation model that may refer to various items, as claimed. The items claimed are script-based and may be used to schedule events.

SRML is a XML based language. So the stated comparison between the XML based languages mentioned in Huang and the simulation model created by SRML is not a proper comparison. Besides the XML based languages mentioned by Huang are not even simulation related and are used in the transport of data over the Web. Huang states that a common standard or Document Type Definition (DTD) using XML may be developed for the transport of simulation data over the Web and refers to areas other than simulation in which such a standard is being used, such as mathematics, chemistry, and biology.

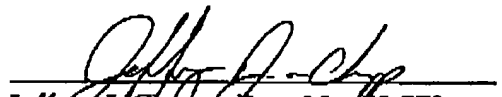
In addition, like Miller '97, Huang also fails to teach or suggest a simulation reference simulator that receives a SRML model, a SRML simulator model, a SRML model that includes a simulation item by way of simulation specific scripts, and a SRML that is domain specific to simulations. Referring to MPEP 706.02(j) and 2143, to establish a *prima facie* case of obviousness the prior art reference(s) must teach or suggest all the claim limitations, see *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). Thus, since Miller '97 and Huang alone or in combination fail to teach or suggest each and every element of claims 1, 9, 17, and 23, Applicants submit that claims 1, 9, 17, and 23 are novel, nonobvious, and are in a condition for allowance. Also, since claims 2-8, 10-16, and 18-21 depend from claims 1, 9, and 17, respectively, they too are novel, nonobvious, and are in a condition for allowance for at least the same reasons.

In light of the amendments and remarks, Applicants submit that all the rejections are now overcome. The Applicants have added no new matter to the application by these amendments. The application is now in condition for allowance and expeditious notice thereof is earnestly solicited. Should the Examiner have any questions or comments, he is respectfully requested to contact the undersigned attorney.

The Commissioner is hereby authorized to charge any fees related to this Office Action response or credit any overpayments to Deposit Account No. 50-0476.

Respectfully submitted,

ARTZ & ARTZ, P.C.



Jeffrey J. Chapp, Reg. No. 50,579
28333 Telegraph Road, Suite 250
Southfield, MI 48034
(248) 223-9500

Dated: August 9, 2005